On the use of Ontology Data for Protecting Critical Infrastructures

João Henriques[1,2], Filipe Caldeira[1,2], Tiago Cruz[1], and Paulo Simões[1]
[1]Department of Informatics Engineering, University of Coimbra, Portugal
[2]Polytechnic Institute of Viseu, Portugal
jpmh@dei.uc.pt
fmanuel@dei.uc.pt
tjcruz@dei.uc.pt
psimoes@dei.uc.pt

Abstract: Modern societies increasingly depend on products and services provided by Critical Infrastructures (CI) in areas such as energy, telecommunications and transportation, which are considered vital for their wellbeing. These CIs usually rely on Industrial Automation and Control Systems (IACS), which are becoming larger and more complex due to the increasing amount of available heterogeneous data generated by a raising number of interconnected control and monitoring devices and involved processes. The Security Information and Event Management (SIEM) systems in charge of protecting these CI usually collect and process data from specialized sources, such as intrusion detection systems, log sources, honeypots, network traffic analysers and process control software. However, they usually integrate only a small fraction of the whole data sources existing in the CI. Valuable generic data sources such as human resources databases, staff check clocks, outsourced service providers and accounting data usually fall outside the specialized perimeter of SIEM, despite their potential usefulness for achieving a truly holistic perspective on the CI cybersecurity awareness. One of the main reasons for this state of affairs is the difficulty of integrating such data into the SIEM systems, since it is usually dispersed across multiple databases, using different schema and not originally intended for security-related applications. The process of collecting and adapting data from each of those sources would typically require a long and expensive process of conversion from each database. Moreover, since these databases may change over time (e.g. introduction of new Human Resources information systems), the system is difficult not only to setup but also to properly maintain over time. In order to address this gap, in this paper we propose a framework for making this process easier by using a semantic web approach for automated collection and processing of corporate data from multiple heterogeneous sources. This way, it becomes possible to make these data available, at reasonable costs, in a format which is suitable for security management purposes – especially those related with audit compliance and forensic analysis.

Keywords: critical infrastructure protection (CIP), security information and event management (SIEM), industrial automation and control systems (IACS), semantic web, ontologies

1. Introduction

Critical Infrastructures (CI) such as telecommunication networks and power grids are becoming increasingly complex and interdependent on people, processes, technologies, information and other critical infrastructures. Operators in charge of Critical Infrastructure Protection (CIP) are required to improve their security levels through different perspectives, including compliance auditing – related with applicable security regulations, standards and best practices – and forensic analysis which typically benefit from assuming a broader scope, beyond the specific operations of the CI industrial control systems and also encompassing other areas of the organization.

The benefits of enlarging the scope of information sources for SIEM applications, forensic analysis and audit compliance operations are rather evident, since this enables more powerful, all-inclusive approaches to cybersecurity awareness. One simple example is the correlation of mail filtering logs (monitoring phishing and malware attacks targeting the employees of the CI), information about employee functions residing in Human Resources information systems and monitoring of abnormal activity within the IACS specific domain. Another example is the correlation of data from physical access control systems and staff check clocks with activity logs of IACS operators. In general, this strategy of associating core security information already fed into SIEM systems with peripheral awareness data results in richer security analysis processes, enabling the detection of inconsistencies, malpractices and intrusion clues which would otherwise go unnoticed.

However, achieving tight integration of all those peripheral data sources into the already existing SIEM frameworks is costly and often unpractical. This would require considerable investments in data conversion and adaptation to the SIEM data flows. Moreover, the maintenance costs would also be considerable, since even minor adjustments on the corporate information systems would require explicit adaptations on the SIEM side.

A more plausible option is therefore the adoption of loosely coupled integration strategies, such as resorting to Semantic Web approaches for automating the processing and interpretation of large amounts of information available from both local databases and Internet repositories. This reasoning process, applied over a large quantity of available data with knowledge inferred from a combination of axioms, properties and rules (e.g., with different levels of hierarchies or categorizations and deriving conclusions) can be explicitly expressed by ontologies.

It should be noted that most data are still not directly available in Semantic Web formats. This is, for instance, the case of data maintained in Relational Databases (RDBs). Nonetheless, mapping data from RDB to Semantic Web-enabled Resource Description Frameworks (RDF) has been the focus of a large body of previous research, leading to the implementation of many generic mapping tools and their application on several specific domains. Those tools are natural candidates to be adapted to the field of CIP, in order to look up for security-related ontology data currently stored in heterogeneous databases – despite the considerable challenges involved, such as the migration from existent systems to the semantic level (Sernadela, P., González-Castro, L., et al 2017).

A detailed discussion of the main motivations and driving research efforts in mapping RDB to RDF can be found in (Sahoo, S., Halb, et al. 2009). Although most models are able to perform inference from native ontology data stores, data still reside mostly in RDBs, which are broadly used within organizations. Moreover, the growing number of datasets published on the Web brings opportunities for extensive data availability and challenges related to the process of querying data in a semantically heterogeneous and distributed environment. The structured query approach fails on the linked data because the Web's scale makes it impracticable for users to know in advance the structure of datasets (Freitas, A., Curry, et al. 2012).

This paper introduces a new approach, considering inference capabilities from Semantic Web, supported by common schemas, for creating a set of independent databases, each deployed with its own domain-specific schema. This kind of reasoning is suitable for application in the context of Critical Infrastructure Protection, and therefore it is able to leverage current SIEM capabilities – mainly in what relates to forensic and audit compliance processes, but also for intrusion detection purposes. This large amount of living heterogeneous data that still reside in the organizational RDBs will, in this way, become available to the Critical Infrastructure's SIEM – enabling new valuable insights into available configuration and monitoring data.

After discussing some of the key previous work and trends in the area, this paper takes a practical approach, presenting the implementation of a federated query architecture for retrieving a set of compliance auditing rules that may be useful, for instance, for assessing CI security levels. In order to leverage inference capabilities, it maps the living data currently available on RDBs into RDFs formats. This way, it is able to substantially enlarge the data available to the SIEM, taking advantage of the large amount of heterogeneous data of production RDB systems. Such an approach provides an abstraction mechanism for maintaining the data consumers away from low-level details, while leveraging the security concerns of the underlying infrastructures by hiding the internal deployment aspects such as the identification of the involved machines and their RDB schemas.

The main goal of this work is the use of an ontology-based approach by considering the available information currently stored in RDB, making it accessible through simple interfaces that collect queried data from multiple natively different data repositories within the organization. Each available RDB maintains different information instances, deployed on specific schemas and technologies. Such an approach is suitable for combining data from two different worlds, such as the case of RDB and semantic web data, which is natively maintained in RDF stores and made available through an interface layer encapsulating the details of the gathering process to retrieve the data from multiple different RDBs.

The remainder of this paper is structured as follows. Section 2 discusses the background for the domain problem and related work. Section 3 analyses the applicability of ontology data in the context of CIP. Section 4 describes the architecture and details its implementation. Finally, Section 5 concludes the paper.

2. Background

This section briefly introduces the reader to the key concepts and tools used in the proposed data integration approach: RDF; RDB and RDF mapping; SPARQL; Direct Mapping of Relational Data to RDF; and the D2RQ platform.

2.1 Resource Description Framework (RDF)

The Resource Description Framework aims at representing information that may be used for inference purposes over the Web. The RDF syntax core structure comprises a set of triples with a subject, a predicate, and an object. A set of triples is called an RDF graph. An RDF graph may be visualized as a directed-arc diagram, in which each triple is represented as a node-arc-node link. RDF is a data format based on a web-scalable architecture for identification and interpretation of terms (RDF 2014).

2.2 Mapping from RDF to RDB

As already mentioned, the mapping of large amounts of data from RDB to RDF has been the focus of intense research work in multiple domains and has led to the implementation of a set of generic mapping tools, as well as domain specific applications. RDF has provided an integration platform for data gathered from multiple sources, primarily from RDB. This is one of the main motivations driving research efforts on mapping RDB to RDF, using various approaches (Seaborne 2013).

SPARQL (W3 2013) can be used to express queries across diverse data sources, whether for data natively stored as RDF or for data viewed as RDF via some sort of middleware. SPARQL is a World Wide Web Consortium (W3C) recommendation for querying multiple RDF graphs. The SPARQL specifications define the syntax and semantics to proceed with queries across diverse stored natively RDF data sources. Using the latest stable release (1.1), SPARQL federated queries allow merging multiple results, retrieved from multiple RDF sources. The syntax and semantics of SPARQL 1.1 Federated Query extension allow distributed queries over different SPARQL endpoints. Moreover, the SERVICE clause extends SPARQL 1.1 to support queries that merge data distributed across the Web. A single query is therefore able to return related data (e.g. contacts to be applied to user John Doe) from multiple distinct SPARQL endpoints.

2.3 Direct mapping of relational data to RDF

Relational databases allow the use of tools such as Structured Query Language (SQL) for accessing and managing the databases. Several strategies already exist in order to map relational data to RDF. Typically, the goal is to describe the RDB contents using an RDF graph, allowing queries submitted to the RDF schema to indirectly retrieve the data stored in relational databases. A direct mapping process enables a simple transformation and can be used for materializing RDF graphs or for defining virtual graphs, which can be queried via SPARQL or traversed by an RDF graph Application programming interface (API). A mapping document is an RDF document containing triples maps with instructions on how to convert relational database content into RDF graphs.

2.4 The D2RQ platform

The D2RQ (Data to RDF Query) Platform allows accessing relational databases as virtual, read-only RDF graphs, automatically producing the corresponding mappings. It is available under the Apache open source license (D2RQ 2012), and it allows users to create customized mappings from RDB through an integrated environment with multiple options for accessing relational data, including RDF dumps, Jena and Sesame API based access, and SPARQL endpoints on D2RQ Server (Bizer, Cyganiak 2007). It offers RDF-based access to the content of RDB without requiring its replication into RDF stores. D2RQ therefore allows querying non-RDF databases using SPARQL or accessing contents of databases over the Web. It also allows the creation of custom content dumps from relational databases into RDF stores.

The D2RQ Platform includes components such as a Mapping Language, an Engine, and a D2R (Data to RDF) Server. The D2RQ Engine is a plug-in for the Jena Semantic Web toolkit, which uses mappings for rewriting the Jena API calls to SQL queries against the database and for redirecting query results up to the higher layers of the framework. The D2R Server is an HTTP server providing linked data views, HTML views for debugging and a SPARQL protocol endpoint providing an interface to query the database. The D2RQ platform supports databases such as MySQL, SQL Server, Oracle, PostgreSQL, HSQLDB and Interbase/Firebird. Some limitations of D2RQ include the integration of multiple databases or other data sources and its read-only nature, lacking create, read, update and delete (CRUD) operations as well. Finally, it does not support inference mechanisms and does not include named graphs (D2RQ 2012).

The D2RQ Mapping Language enables defining relationships between RDB schemas and RDF schema vocabularies (classes and properties) or Web Ontology Language (OWL) ontologies written in Turtle syntax (W3 2014). The mapping properties define a virtual RDF graph, which contains information from the database schema. The mapping process between D2RQ and RDB entities includes the RDF class node to RDB tables and RDF predicates to RDB column names (D2RQ 2012).

3. Applicability of ontology data in the context of critical infrastructure protection

In this section we address the applicability of ontology data in the context of CIP. First, we discuss some of the more relevant related works. Afterwards, we present the H2020 ATENA module for forensics and audit compliance that provides the framework on which we developed our proposed approach – which is next described, in Section 4.

3.1 Related work

Current approaches on the use of ontologies in the context of CIP are mostly related with the assessment of interdependencies between Critical Infrastructures, such as the works of Castorini et al. (Castorini, Palazzari, et al. 2010) and Blackwell et al. (Blackwell, Tolone, et al. 2008). Similarly, a proposal for an ontology providing vulnerabilities classification to be used in decision support tools can be found in (Chora´s, Kozik, et al. 2010).

Other worth mentioning approaches include SPLENDID, DARQ, SemaPlorer and FedX. SPLENDID (Gorlitz, Staab 2011) is a query optimization strategy for federating SPARQL endpoints based on statistical data. DARQ (Quilitz 2008) provides transparent query access to multiple SPARQL services through the use of one single RDF graph, even when data have a distributed nature and is spread over the web. This approach includes a service description language that enables a query engine to decompose a query into subqueries, where each of them can be answered by an individual service. SemaPlorer (Schenk, Saathoff, et al. 2009) also provides a federated query architecture allowing to interactively explore and visualize semantically heterogeneous distributed semantic datasets in real-time, through a conceptual layer on top of Amazon's Elastic Computing Cloud (EC2). FedX (Schwart, Haase, et al. 2011) proposes novel joint processing and grouping techniques for minimizing the number of remote requests. It also develops a practical framework that enables efficient SPARQL queries supported by federation layers for efficient query processing on heterogeneous distributed Linked Open Data sources.

As already mentioned, one possible application of ontology data in this scope is the usage of heterogeneous sources available in organizational RDBs for leveraging inference capabilities. This is especially interesting in the specific areas of forensic analysis and audit compliance processes, which by nature need to be supported by substantial amounts of heterogeneous data. A possible practical application of this approach, in the scope of forensic analysis and audit compliance processes, may consist on the collection and mapping to Semantic Web of rules residing in the multiple and heterogeneous relational databases of the CI organization – so they can be combined with the knowledge already available at the SIEM systems. This path has been explored in the scope of the H2020 ATENA research project (ATENA 2018)(Rosa, Proença et al. 2017), as discussed next.

3.2 Forensics and compliance auditing in the scope of the H2020 ATENA framework

The H2020 ATENA project proposes an innovative logical framework with design improvements of role, operation, architecture, and security components for Industrial Automation and Control Systems (IACS), while also exploiting novel security approaches enabled by network softwarization paradigms. The Forensics and Compliance Auditing (FCA) module, integrated into the ATENA cyber-security architecture, addresses the gathering and persistent storage of digital evidence retrieved from both the cyber-analysis layer (e.g. the SIEM) and peripheral sources, such as service logs, sessions or physical access control systems, among others – for forensics and compliance auditing purposes. Its forensics tools provide the means to identify, extract, preserve, and highlight digital evidence for technical investigation and legal purposes. Its compliance auditing tools support the audit procedures associated with certification processes for applicable standards, policies and regulations – for example, verifying the authorization procedures for physical installation access, such as access to doors (Rosa, Proença et al. 2017).

Moreover, the FCA module provides a set of analysis capabilities for interactively exploring, searching, extracting, pinpointing and combining insights from available data. The core FCA functions encompass collecting heterogeneous data from internal and external sources, producing structured and unstructured data to be combined and gathered into a unified view for auditing compliance – throughout a set of rules – and also providing forensic investigation functionalities for retrieving evidence.

Figure 1 depicts the main blocks of the ATENA FCA module. Data collected from the ATENA SIEM and intrusion detection systems feed a specific CI security data lake which provides input to the FCA analytics components. Peripheral data sources, processed through domain-specific business rules, also feed the analytics layer. Trust and repudiation indicators are also use, to assess the trustworthiness of each data source.

As previously discussed, specific ontologies need to be constructed for supporting the already mentioned processes of audit compliance and forensics analysis. In the context of the FCA module hereby presented, the targets for the use of those ontologies are the Analytics sub-components “Audit Compliance” and "Forensic Analysis”.

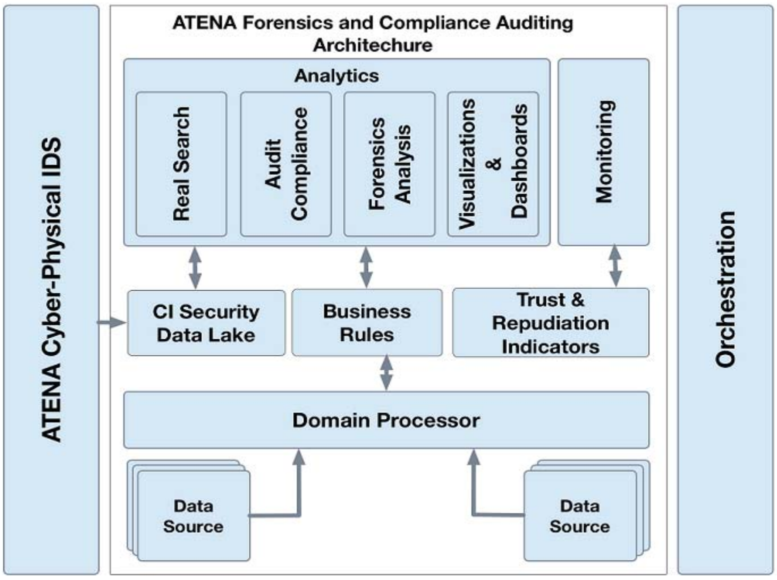

Figure 1: The forensics and audit compliance module of the ATENA Project (Rosa, Proença et al. 2017)

4. Proposed approach to the use of ontology data for CIP

This section describes the proposed approach to the use of ontology data in the context of CIP applications. First, we introduce the proposed reference architecture, followed by the discussion of technical aspects and implementation details. In a simplified view, the proposed solution consists in a web service that is able to receive several SPARQL requests from data consumers (such as the forensics and audit compliance tools mentioned in the previous sections). Afterwards, each one of those requests is forwarded into different databases deployed using different schemas.

4.1 Reference architecture

The proposed reference architecture, depicted in Figure 2, comprises a set of components such as: a **federated layer**; **mapping brokers;** and **databases**. Several data consumers (clients) may send distinct sets of SPARQL queries to the federated interface layer, which delivers each query to all the brokers. The broker main role is to transform the incoming SPARQL queries into native relational database queries. Through an inverse flow, the broker retrieves the data subsets from the database to be gathered into full data sets at the federated layer and finally forwarded to the involved client(s).

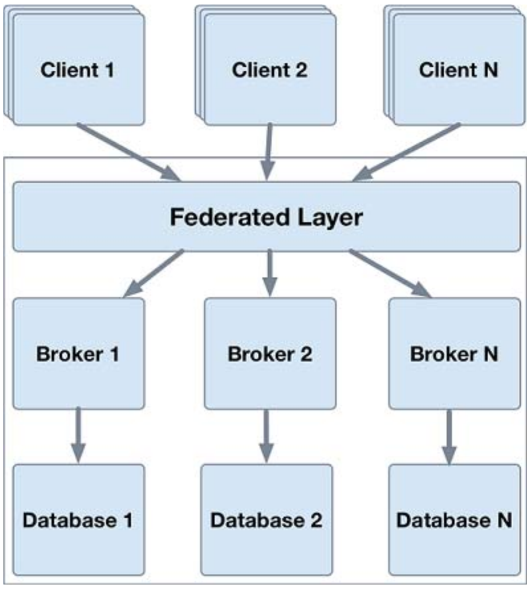

Figure 2: Proposed reference architecture

4.2 Implementation aspects

The interface layer is implemented as a web service, while the mapping brokers are implemented as D2R Server endpoints. Each endpoint is assigned to different relational database(s)**.** Figure 3 provides a general overview of the implementation of the described architecture, depicting how requests flow from a submitted query to the web service, which implements a federated query solution to dispatch the incoming requests to the indexed list of database servers – with each of them mapped by a specific D2RQ component. For sake of simplicity, the figure includes just two different databases with different schemas (one Microsoft database – MSSQL – and one MySQL database), but there are no limits to the number or type of involved databases. The use case hereby described involves a client requesting the contents of the “Rules” database entity. The objective is to gather and combine – without requiring he end user to be aware of low level details – information dispersed across different tables, different databases and using different schemas. After the request query to retrieve the existing contents from the “Rules” entity has reached the database instances, each of them delivers its contents to a SPARQL endpoint through a D2R server assigned to each involved database. The D2RQ Mapping language is used for the mapping process. This central web service allows its clients to directly query existing entities, retrieving available contents from each existing database and finally merging and delivering them to the querying clients.

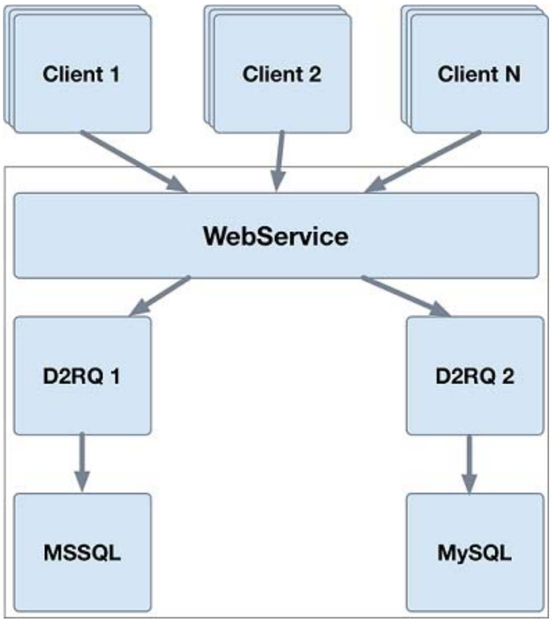

Figure 3: Architecture implementation

Required tools and technologies include Visual Studio as development environment, C# as programming language, ASPX.NET for implementing the web service, classic RDBs such as MSSQL and MySQL, and the RDF and SPARQL languages describing their semantics. Next, we discuss some details for each step involved in the implementation and deployment of this specific use case:

- **MSSQL deployment*:*** this step creates a database and a table in the MSSQL Server, including also the statements for populating it. For evaluation purposes, the contents in the table entity are different from those stored in the MySQL database deploying the same entity table.
- **MySQL server deployment:** this step involves the creation of a table in a MySQL database and the execution of statements for populating it. For sake of demonstration, the schemas and contents of this table are different from those used in the MSSQL database. At the end, these two databases should maintain different data over distinct schemas, which will become federated at the upper level of the web service.
- **D2R to MySQL:** a generate-mapping command produces the mapping file containing the required information in the mapping process from the RDB MySQL Server tables and columns to the RDF model. The following command generates the mapping file from parameters such as user, password and the location of the database:

```
generate-mapping -u root –p password02pt -o ssfile_mysql.ttl jdbc:mysql://server02pt:3307world
```

- **MSSQL server mapping:** the mapping process between the information available in the MSSQL server and RDF is mapped through a SS file named *SQLSERVER.ttl*. The contents of that file map data between RDF graphs and RDB. Properties in this file include the mapping between the MSSQL server schema and RDF graphs. The initial section of this file provides a set of prefixes, removed from this listing. The component *map:database* provides a way to retrieving information from the MSSQL Server *server01pt* and the database named *BD server01pt*.
- **MySQL mapping:** these steps are similar to the MSSQL server mapping, differing only on the involved target databases (in this case is MySQL database). The SS file *SQLSERVER.ttl* provides the mapping mechanism between RDF and the MySQL relational database. This file contains the components, manually updated, providing the correct mapping to the MySQL database, according to its internal schema.
- **D2R Server to MSSQL:** the following command uses the D2R generate-mapping tool to generate the mapping file containing the properties for mapping of tables and columns between RDB MSSQL Server and the RDF model. For this purpose, the following command is used (including parameters such as user, password and location of the MSSQL Server database):

```
generate-mapping -u sa –p password01pt -o ssfile_SQLServer.ttl –d \
com.microsoft.sqlserver.jdbc.SQLServerDriver
```

- **D2R server with the MySQL server mapping:** the next step performs the deployment of the D2R server on port 2021, supported by the previously created mapping SS *SQLSERVER.ttl*. This file contains the information for mapping information from RDF to the MySQL database MySQL. The file *ssfile_SQLSERVER.ttl* provides the mapping mechanisms between RDF and MySQL relational databases, according to its internal schema. The following command was used:

```
d2r-server -p 2021 ssfile_SQLSERVER.ttl
```

- **D2R Server with the MSSQL Server Mapping:** the next step performs the deployment process for the D2R server, specifying its internal port (2020, in this case) and mapping the contents to be retrieved from RDB to RDF according to the mapping file *ssfile_MySQL.ttl*. The following command was used:

```
d2r-server -p 2020 ssfile_MySQL.ttl
```

- **D2R Server with the MySQL Server Mapping:** the next step deploys the D2R server on port 2021, supported in the previously created mapping file *ssfile_SQLSERVER.ttl*. This file contains mapping information from RDF to the database MySQL. The following command was used:

```
d2r-server -p 2021 ssfile_SQLSERVER.ttl
```

- **Web Service:** the web service provides the main functions performing the federation mechanism and retrieving the information from the SPARQL endpoints. The web service provides an interface and a federated query layer offering query services that allow end users to perform the intended inference operations. This approach allows those clients to be abstracted from low-level details. Each submitted is forwarded to multiple RDBs through a DR2Q component, and results are later merged into a single result set. The endpoints are parameters configured at server level, taking in consideration that the end user doesn't need to know nor the number nor the location of such existing endpoint servers. The purpose of the following query is to retrieve the ”Rules” from the different RDB systems:

```
SELECT DISTINCT * WHERE { ?s ?p ?o; vocab:dbo_Rules_RNumber?RNumber; vocab:dbo_Rules_RName
\ ?RName FILTER(?p = vocab:dbo_Rules_IDRule) }
```

5. Discussion and conclusions

In this paper we proposed an approach for leveraging inference capabilities in the use of heterogeneous data currently maintained in multiple, natively different RDB systems. This approach aims at contributing to Critical Infrastructure Protection by supporting activities such as forensic analysis and audit compliance procedures. It provides semantic web reasoning capabilities through an interface able to answer to federated queries. The process of interactively exploring, searching, extracting, pinpointing and combining insights can use and combine data sourced from disparate organizational RDBs. Thus, this approach avoids the duplication of information in RDB and RDF stores, and overcomes the issues arising from the use of static data integration (such as the lack of support for transformations of data and the effort required for maintaining up to date synchronization processes). The proposed web service includes an abstraction layer that deals with inherent complexities of resorting to different platforms, systems, technologies and information schemas to retrieve and combine heterogeneous data. This abstraction layer also improves security by hiding the infrastructure internal details.

Although the approach taken by the proposed federated architecture is similar to the one of SPARQL 1.1, it does not require previous knowledge about the existence and location of SPARQL endpoints. The benefits of this approach come from the inclusion of an abstraction layer providing direct access to operational data that live in different organizational RDBs. Details such as the involved database servers and differences between schemas can be kept away from users. Moreover, it is flexible enough for leveraging the exploration of additional data sources that might be easily added in the future. The proposed framework also provides a data fusion solution for gathering multiple data – representing the same real-world object – into a single, consistent and clean representation.

The need that led to this work arises from the limited research on the use of ontology data for CIP applications, and the need to improve and facilitate the usage of the huge amounts of data living in the RDBs of Critical Infrastructure operators. This work also explored semantic web inference tools aiming at the practical objective of federating queries to a dataset containing “Rules” maintained along different RDBs. This practical approach suggests a future path for the improvement of CIP, supported by the use of inference capabilities for forensic and audit compliance purposes, leveraging the use of heterogeneous ontology data living in RDBs.

Acknowledgements

This work was partially funded by the ATENA H2020 Project (H2020-DS-2015-1 Project 700581).

References

ATENA (2018), H2020 ATENA Project website, [online], https://www.atena-h2020.eu/

Bizer, C., Cyganiak, R. (2007) D2RQ. Lessons Learned. Position paper for the W3C Workshop on RDF Access to Relational Databases, Cambridge, USA.

Blackwell, J., Tolone, W. J., Lee, S. W., Xiang, W. N., and Marsh, L. (2008) An ontology-based approach to blind spot revelation in critical infrastructure protection planning. In International Workshop on Critical Information Infrastructures Security (pp. 352-359). Springer, Berlin, Heidelberg.

Castorini, E., Palazzari, P., Tofani, A., & Servillo, P. (2010) Ontological framework to model Critical Infrastructures and their interdependencies. In Complexity in Engineering, COMPENG'10. (pp. 91-93).

Chora ´s, M., Kozik, R., Flizikowski, A., and Hołubowicz, W. (2010) Ontology applied in decision support system for critical infrastructures protection. Trends in Applied Intelligent Systems, 671-680.

D2RQ (2012), D2RQ, [online], http://d2rq.org.

Freitas, A., Curry, E., Oliveira, J. G., and O'Riain, S. (2012) Querying heterogeneous datasets on the linked data web: challenges, approaches, and trends. IEEE Internet Computing, 16(1), 24-33.

Gorlitz, O., and Staab, S. (2011) Splendid: Sparql endpoint federation exploiting void descriptions. In Proceedings of the Second International Conference on Consuming Linked Data Volume 782 (pp. 13-24). CEUR-WS.org.

Quilitz B. and Leser U. (2008) Querying Distributed RDF Data Sources with SPARQL. In: Bechhofer S., Hauswirth M., Hoffmann J., Koubarakis M. (eds) The Semantic Web: Research and Applications. ESWC 2008. Lecture Notes in Computer Science, vol. 5021. Springer, Berlin, Heidelberg.

RDF (2014), W3C Resource Description Framework (RDF), [online], https://www.w3.org/RDF

Rosa L., Proença J., Henriques J., Graveto V., Cruz T., Simões P., Caldeira F. and Monteiro E. (2017) An evolved security architecture for distributed Industrial Automation and Control Systems,16th European Conference on Cyber Warfare and Security (ECCWS).

Sahoo, S., Halb, W., Hellmann, S., Idehen, K., Thibodeau, T., Auer, S., Sequeda, J., Ezzat, A. (2009) A Survey of Current Approaches for Mapping of Relational Databases to RDF, [online], http://www.w3.org/2005/Incubator/rdb2rdf/RDB2RDF SurveyReport.pdf.

Seaborne A.,Polleres A.,Feigenbaum L. and Williams G. (2013) SPARQL 1.1 Federated Query Position paper for the W3C Workshop on SPARQL 1.1 Federated Query.

Schenk, S., Saathoff, C., Staab, S., and Scherp, A. (2009) SemaPlorer—Interactive semantic exploration of data and media based on a federated cloud infrastructure. Web Semantics: Science, Services and Agents on the World Wide Web, 7(4), 298–304. http://doi.org/10.1016/j.websem.2009.09.006.

Sernadela, P., González-Castro, L., Oliveira, J. L. (2017) SCALEUS: Semantic Web Services Integration for Biomedical Applications. Journal of Medical Systems, 41(4), 54.

W3 (2013), SPARQL, [online], https://www.w3.org/TR/sparql11-query

W3 (2014), SPARQL, [online], https://www.w3.org/TR/turtle